

## 1. IDENTIFICAÇÃO

Padrão **Desenvolvimento de WebApps e App(mobile)**

Seguimento **Arquitetura de Soluções**

Código **P05.008**

Revisão **v. 2023**

## 2. PUBLICAÇÃO

| Versão  | Data para adoção      | Publicação                                     |
|---------|-----------------------|--|
| v. 2017 | 02 de janeiro de 2017 | PORTARIA “N” Nº 258 de 30 de dezembro de 2016. |
| v. 2023 | 05 de junho de 2023   |  |

## 3. PROPÓSITO DO PADRÃO

A padronização para o desenvolvimento de aplicações do tipo *WebApp* e *App(mobile)* visando uniformizar as aplicações desenvolvidas por equipes internas ou fornecedores, permitindo uma clara definição das práticas, técnicas, padrões, ferramentas e frameworks utilizados na IplanRIO, possibilitando:

- Empacotamento de Aplicativos Híbridos para Dispositivos Móveis;
- Melhora da Qualidade Interna e Externa das aplicações;
- Aumento da Produtividade;
- Otimização das Manutenções Evolutivas e Corretivas;
- Melhoria no Intercâmbio de desenvolvedores entre Projetos;
- Aumento do Reuso de Componentes;
- *Mentoria* de Equipes inexperientes;
- Automação de tarefas sujeitas a falhas realizadas pelo Desenvolvedor;
- Melhor Integração dos *Softwares* com a infraestrutura do *Data Center*;
- Repositório de exemplos de códigos e boas práticas de desenvolvimento;
- Estabelecer o uso com plataformas de gerenciamento de contêineres;
- Integração entre aplicações.

A definição do padrão permite a criação de uma norma que atenda a maiorias das necessidades no desenvolvimento de *WebApp* e *App(mobile)* na IplanRIO, e que possa ser distribuída e utilizada por todos os desenvolvedores.

## 4. RESPONSÁVEL PELO PADRÃO

|           |  |
|-----------|--|
| Órgão     | IplanRIO                                   |
| Diretoria | DSI – Diretoria de Sistemas                |
| Setor     | GAI – Gerência de Arquitetura e Integração |
| Contato   | Gerente da GAI                             |

## 5. DESCRIÇÃO DO PADRÃO

**WebApps** são softwares responsivos projetados para utilização através de um navegador, utilizando tecnologias web HTML, JavaScript e CSS. Podem ser executados localmente, a partir de um servidor HTTP (Web Host) ou no dispositivo do usuário.

**App(mobile)** são softwares desenvolvidos para dispositivos móveis, como tablets e smartphones. Eles podem ser baixados em lojas virtuais, como a Apple App Store e a Google Play Store.

Este padrão reúne um conjunto de especificações, processos, ferramentas e *frameworks* a serem adotados para a construção de **WebApps** e **App(mobile)** englobando as funcionalidades de documentação e a automatização de alguns processos que auxiliam as fases de desenvolvimento e homologação das aplicações.

## 6. POLÍTICA E NORMATIZAÇÃO DE USO

- 6.1. Fica estabelecido como padrão tecnológico para **Desenvolvimento de WebApps e App(mobile)**, os itens especificados no **tópico 7**.
- 6.2. O desenvolvimento e/ou aquisição de novas soluções sistêmicas deverão estar de acordo e seguir a referência recomendada listada na especificação técnica deste padrão.
  - 6.2.1. A aquisição de novas soluções deverá ter a IplanRIO como **RESPONSÁVEL TÉCNICA**;
  - 6.2.2. A **RESPONSÁVEL TÉCNICA** deve ser comunicada antecipadamente sempre que for necessário custear o licenciamento de qualquer componente de software pertencente à solução.
- 6.3. Todas as exceções e dúvidas relacionadas a este documento devem ser tratadas com o **responsável pelo padrão**;
- 6.4. Com o objetivo de atualização, modernização e capacidade de melhor atender as demandas, os componentes do padrão tecnológico **Plataformas Web** serão revistos pela **Diretoria de Sistemas da IplanRIO** e pelo responsável pelo padrão.
- 6.5. Todas as exceções e dúvidas relacionadas a este documento devem ser tratadas

com o responsável pelo padrão;

## 7. ESPECIFICAÇÃO TÉCNICA

### 7.1. Especificação dos componentes WebApps (Front-end):

| Componente                        | Especificação  | Situação |
|-----------------------------------|--|----------|
| <i>Runtime</i>                    | Node JS (Interpretador)  | Adotado  |
| <i>Linguagem para Codificação</i> | React JS (Framework JavaScript)  | Adotado  |
| Bibliotecas                       | Redux/ReduxToolKit (Controlador de fluxo e estados)  | Adotado  |
| Autenticação                      | via <a href="#">Gov.br</a><br>Roteiro: <a href="#">Clique aqui!</a>  | Adotado  |
| <i>Paginas Web</i>                | Modelo de <i>Single Page Application-SPA</i> para o desenvolvimento de aplicações web responsivas e aplicativos móveis híbridos utilizando HTML5 e consumindo serviços construídos na arquitetura REST.  | Adotado  |
| <i>Web Components</i>             | Conjunto de recursos atualmente adicionados pelo W3C às especificações HTML e DOM que permitem a criação de widgets ou componentes reutilizáveis em documentos web e aplicativos da web.   | Adotado  |
| <i>Padrão Arquitetural</i>        | Microserviços  | Adotado  |
| Plataforma                        | Docker (Openshift)   | Adotado  |
| <i>Modelagem de Software</i>      | Adotar Design Patterns e os princípios S.O.L.I.D. para a gestão de dependências dos objetos e componentes:<br><br>S - Single Responsibility Principle (Princípio de Responsabilidade Única);<br>O - Open Close Principle (Princípio Aberto - fechado);<br>L - Liskov Substitution Principle (Princípio de Substituição de Liskov);<br>I - Interface Segregation Principle (Princípio de Segregação de Interface);<br>D - Dependency Inversion Principle (Princípio Inversão de dependência). | Adotado  |

## 7.2. Especificação dos componentes WebApps (Back-end/API):

| <b>Componente</b>                 | <b>Especificação</b>  | <b>Situação</b> |
|-----------------------------------|---|-----------------|
| <i>Runtime</i>                    | JVM (Interpretador), Node JS (Interpretador)                        | Adotado         |
| <i>Linguagem para Codificação</i> | Quarkus (Java), Node JS   | Adotado         |
| <i>Bibliotecas</i>                | Hibernate, REStEasy, Vert.x e MicroProfile                          | Adotado         |
| <i>Autorização</i>                | via <a href="#">Gov.br</a><br>Roteiro: <a href="#">Clique aqui!</a> | Adotado         |
| <i>Padrão Arquitetural</i>        | Microserviços   | Adotado         |
| <i>Plataforma</i>                 | Docker (Openshift)  | Adotado         |

### 7.3. Especificação dos componentes App(mobile):

| Componente                        | Especificação  | Situação |
|-----------------------------------|--|----------|
| <i>Runtime</i>                    | Node JS (Interpretador)  | Adotado  |
| <i>Linguagem para Codificação</i> | React Native (Framework JavaScript)  | Adotado  |
| <i>Bibliotecas</i>                | Redux/ReduxToolKit (Controlador de fluxo e estados)  | Adotado  |
| <i>Autenticação</i>               | via <a href="#">Gov.br</a><br>Roteiro: <a href="#">Clique aqui!</a>  | Adotado  |
| <i>Padrão Arquitetural</i>        | Microserviços  | Adotado  |
| <i>Plataforma</i>                 | Android, iOS   | Adotado  |
| <i>Modelagem de Software</i>      | Adotar Design Patterns e os princípios S.O.L.I.D. para a gestão de dependências dos objetos e componentes:<br><br>S - Single Responsibility Principle (Princípio de Responsabilidade Única);<br>O - Open Close Principle (Princípio Aberto - fechado);<br>L - Liskov Substitution Principle (Princípio de Substituição de Liskov);<br>I - Interface Segregation Principle (Princípio de Segregação de Interface);<br>D - Dependency Inversion Principle (Princípio Inversão de dependência). | Adotado  |

## 8. DEFINIÇÕES E ABREVIATÓES

| Termo              | Definição   |
|--------------------|---|
| <b>Android</b>     | <b>Android</b> é um sistema operacional (SO) baseado no núcleo Linux e atualmente desenvolvido pela empresa de tecnologia Google. Com uma interface de usuário baseada na manipulação direta, o Android é projetado principalmente para dispositivos móveis com tela sensível ao toque como smartphones e tablets; com interface específica para TV (Android TV), carro (Android Auto) e relógio de pulso (Android Wear). O sistema operacional utiliza-se da tela sensível ao toque para que o usuário possa manipular objetos virtuais e também de um teclado virtual. Apesar de ser principalmente utilizado em dispositivos com tela sensível ao toque, também é utilizado em consoles de videogames, câmeras digitais, computadores e outros dispositivos eletrônicos. |
| <b>Back-end</b>    | <b>Back-end</b> é a camada de uma aplicação responsável pela lógica de negócio onde é realizado o gerenciamento e processamento e armazenamento das informações provenientes do Front-end.  |
| <b>Contêineres</b> | <b>Contêineres</b> se refere a tecnologia de virtualização que propicia a criação e execução de aplicativos de forma isolada em um ambiente virtualizado com todas as suas dependências e bibliotecas em um único objeto, chamado de imagem do contêiner. Podem ser gerenciados de forma centralizada, permitindo a escalabilidade horizontal e vertical de aplicativos e serviços.   |
| <b>Data Center</b> | <b>Data Center</b> é o local onde são concentrados os equipamentos de processamento e armazenamento de dados de uma empresa ou organização.   |
| <b>Framework</b>   | <b>Framework</b> é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação.  |
| <b>Front-End</b>   | <b>Front-End</b> é a camada de uma aplicação responsável por coletar a entrada do usuário em várias formas e processá-la para adequá-la a uma especificação em que o back-end possa utilizar.   |

| Termo             | Definição   |
|-------------------|---|
| <b>HTML5</b>      | <p><b>HTML5 (Hypertext Markup Language, versão 5)</b> é uma linguagem para estruturação e apresentação de conteúdo para a World Wide Web e é uma tecnologia chave da Internet originalmente proposto por Opera Software. É a quinta versão da linguagem HTML. Esta nova versão traz consigo importantes mudanças quanto ao papel do HTML no mundo da Web, através de novas funcionalidades como semântica e acessibilidade. Possibilita o uso de novos recursos antes possíveis apenas com a aplicação de outras tecnologias. Sua essência tem sido melhorar a linguagem com o suporte para as mais recentes multimídias, enquanto a mantém facilmente legível por seres humanos e consistentemente compreendida por computadores e outros dispositivos (navegadores, parsers etc). O HTML5 será o novo padrão para HTML, XHTML, e HTML DOM. Atualmente, está em fase de esboço, porém diversos navegadores já implementam algumas de suas funcionalidades.</p> |
| <b>IOS</b>        | <p><b>iOS (antes chamado de iPhone OS)</b> é um sistema operacional móvel da Apple Inc. desenvolvido originalmente para o iPhone, também é usado em iPod touch e iPad. A Apple não permite que o iOS seja executado em hardware de terceiros.</p>   |
| <b>Javascript</b> | <p><b>JavaScript</b> é uma linguagem de programação interpretada. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.</p>   |

| Termo                              | Definição  |
|------------------------------------|--|
| <b>Node JS</b>                     | <b>Node JS</b> é um open-source, multiplataforma JavaScript runtime ambiente para o desenvolvimento de uma variedade diversificada de ferramentas e aplicações. Embora Node.js não seja uma estrutura JavaScript, muitos de seus módulos básicos são escritos em JavaScript, e os desenvolvedores podem escrever novos módulos em JavaScript. O ambiente de tempo de execução interpreta o JavaScript usando o mecanismo de JavaScript V8 do Google.   |
| <b>Open-Source</b>                 | <b>Código aberto</b> , ou <b>open source</b> em inglês é um modelo de desenvolvimento que promove um licenciamento livre para o design ou esquematização de um produto, e a redistribuição universal desse design ou esquema, dando a possibilidade para que qualquer um consulte, examine ou modifique o produto.   |
| <b>REST</b>                        | <b>Representational State Transfer (REST)</b> é uma abstração da arquitetura da World Wide Web (Web), um estilo arquitetural que consiste de um conjunto coordenado de restrições arquiteturais aplicadas a componentes, conectores e elementos de dados dentro de um sistema de hipermídia distribuído. O REST ignora os detalhes da implementação de componente e a sintaxe de protocolo com o objetivo de focar nos papéis dos componentes, nas restrições sobre sua interação com outros componentes e na sua interpretação de elementos de dados significantes. |
| <b>Single Page Application-SPA</b> | <b>Um aplicativo de uma única página (SPA)</b> é um aplicativo da web ou site que se encaixa em uma única página da Web com o objetivo de fornecer uma experiência semelhante à de um aplicativo de desktop. Em um SPA, todos os códigos necessários - HTML, JavaScript e CSS - são recuperados com um único carregamento de página, ou os recursos apropriados são carregados dinamicamente e adicionados à página conforme necessário, geralmente em resposta às ações do usuário.   |
| <b>Software</b>                    | <b>Software</b> é uma sequência de instruções a serem seguidas e/ou executadas, na manipulação, redirecionamento ou modificação de um dado/informação ou acontecimento.  |

## 9. REFERÊNCIAS

Wikipedia. Disponível em: <http://pt.wikipedia.org/> e <http://en.wikipedia.org/>. Acesso em 7 de Dezembro de 2016.

## 10. GRUPO TÉCNICO RESPONSÁVEL PELA ELABORAÇÃO DO PADRÃO

Diretoria de Sistemas da IplanRIO  
**Edwilson da Silva**  
**Ricardo Lúcio Gonçalves de Sá**  
**Vladimir Cavalcante Donadio**